TECHNICAL APPLICATION NOTE

– NFC Forum Mandated Type 1 Tag Format

Guide to Tag Programming and Locking:

Topaz 13.56MHz Near Field Communication (NFC) / Radio Frequency Identification (RFID) Read/Write IC

(ISO/IEC 18092, 21481 & 14443A Compatible)

www.innovision-group.com/topaz

INNOVISION RESEARCH & TECHNOLOGY PLC

M2000-1109-01
Revision 1.0 October 2007

Table of Contents

Datasheet Revision History

Revision	Date	Page(s)	Description
1.0	10/07		1st release

UK Head Office

Innovision Research & Technology plc
33 Sheep Street
Cirencester
Gloucestershire
GL7 1RQ, UK

Telephone: +44 (0) 1285 888200
Fax: +44 (0) 1285 888190

Email: topaz@innovision-group.com
Web: www.innovision-group.com/topaz

1. Introduction

The Topaz IC, (part number TPZ-201-series), developed by Innovision Research & Technology plc, has been mandated by the NFC Forum as the Type 1 Tag Format, to address Near Field Communication (NFC) and Radio Frequency Identification (RFID) tagging applications working to the ISO/IEC18092, ISO/IEC21481 and ISO/IEC14443A standards.

Topaz is targeted for operation with Near Field Communications (NFC) devices when operating in Reader/Writer mode.

The Topaz IC is a two terminal device designed to be connected to a loop antenna to produce a passive NFC/RFID tag operating in the standard unlicensed 13.56MHz frequency band.

Passive operation means that no battery is required because the Topaz IC gathers its operational energy from the interrogation field generated by the Reader/Writer unit.

The Topaz IC is based on a physical EEPROM array size of 120 bytes and features 96 bytes of user read/write memory. These are arranged as blocks of 8-bytes, allowing individual blocks to be locked into read only operation by contactless command. Once locked, the process is irreversible.

2. Purpose of this Application Note

There are NFC devices now emerging onto the market, eg Nokia 6131 NFC phone and people are setting up trials, schemes and applications using the Type 1 tag format, such as ‘Smartposter’ and ‘Tags in Box‘ etc.

The purpose of this application note is to help explain and offer guidance for the programming, locking and operation of the Type 1 tag.

It also explains about using an RFID reader to change the type 1 tag to be a *Read-only* operation, ie prevent any future writing of new data.

The typical Smartposter application requires the tag to be programmed with data and then ‘locked’ into a *Read-only* mode such that nobody can either accidentally or maliciously ever change the data again after the Smartposter has been deployed.

The Type 1 tag based on Topaz IC has the capability to be physically locked so that it is irreversibly locked into a *Read-only* condition.

3. References & Related Documents

Ref [1] – Type 1 Tag Operation Technical Specification (NFCForum-TS-Type-1-Tag-1.0)

Ref [2] – Topaz IC Full Technical Datasheet non-NDA M2000-1057-02

4. Tag Programming and Locking

4.1 Programming

Obviously the Nokia 6131 NFC phone can be used for encoding the Bookmark (URL), Text Message (SMS) or Call Request (Phone number) data and then 'sending', ie programming or writing this data into the tag. See Nokia 6131 NFC handbook for instructions.

However, this is only really suited to low quantity of tags and the standard application software currently available on this phone does not offer the facility to 'lock' the data into the tag by making the tag have a *Read-only* capability.

4.2 Innovision Topaz Demonstration Software

Innovision can supply a software application with a GUI for programming data and locking tags by hand. This software GUI operates with an industry standard RFID reader/writer unit and allows for easy programming and duplication of tags as well as performing the tag locking function.

4.3 Commercial Tools

Data encoding and tag programming tools are also commercially available; eg NFC Tag Manager from Alvin Systems:

http://www.alvinsystems.com/resources/pdf/NFCTagManager.pdf

5. NDEF Management

5.1 NDEF

The NFC Forum has defined data formats for the popular use cases of NFC tags. The generic name for this is NFC Data Exchange Format (NDEF). All application data eg Bookmark (URL), Text Message (SMS), Call Request (Phone number), requires encoding into NDEF format before writing or programming into a tag.

5.2 Identification as NFC Forum Type 1 Tag

The memory map of the Type 1 tag has a fixed Header ROM byte called HR0.

To identify the Type 1 tag the high nibble of HR0 is fixed equal to 0001b.

5.3 Write Permission

An NFC Forum compliant device in NFC Forum Reader/Writer Mode should not even attempt to write to a tag unless it is sure that it is a Type 1 tag as confirmed by HR0 = $1x_h$. This pre-qualification is required to protect against accidental writing and corruption of a non-NDEF application tag such as a transit ticket based on an IC operating with the same proprietary protocol, eg Innovision Jewel IC, which has a different HR0 value.

5.4 Confirmation of Presence of NDEF Message

Although the qualification of the HR0 value will have identified the tag encountered as a Type 1 tag and hence capable of carrying an NDEF message, there may or may not be an actual NDEF message present.

So that an NFC Device can quickly recognise the status of the tag, the first four bytes of memory are allocated for the function of a Capability Container (CC).

5.5 Capability Container

The four bytes allocated for the CC are highlighted on the memory map representation in Figure 1 and labelled as CC0 – CC3

- CC0 when equal to $E1_h$, the NDEF magic number (NMN), indicates that NFC Forum defined NDEF message data is stored in the data area
- CC1 is used to carry the Version Number (VNo) of the Tag Operating document. This provides for future upgrade capability
- CC 2 indicates the physical tag memory size (TMS) of the Type 1 tag. This provides for automatic operation with larger memory size versions of Topaz
- CC 3 is used to indicate the read and write access (RWA) capability of the CC and data area of the Type 1 tag

Figure 1 shows an example coding of the CC bytes in a Type 1 tag:

- With NFC Forum defined data (byte 0 = $E1_h$),
- Supporting the version 1.0 of the tag operation mapping document (byte 1 = 10_h),
- With 120 bytes of memory size (byte 2 = $0E_h$),
- With read and write full access granted (byte 3 = 00_h).

HR0	HR1
11_h	XX_h

EEPROM Memory Map									
Memory Block	Byte-0 **(LSB)**	Byte-1	Byte-2	Byte-3	Byte-4	Byte-5	Byte-6	Byte-7 **(MSB)**	Lock Condition
0	UID-0	UID-1	UID-2	UID-3	UID-4	UID-5	25_h		Locked
1	CC0 (NMN) $=E1_h$	CC1 (Vno) $=10_h$	CC2 (TMS) $=0E_h$	CC3 (RWA) $=00_h$	NDEF Message TLV (Tag) $T=03_h$	NDEF Message TLV (Length) $L=00_h$			
2									
3									
4									
5									
6									
7									
8									
9									
A									
B									
C									
D									Locked
E	LOCK-0 $=01_h$	LOCK-1 $=06_h$	OTP-0	OTP-1	OTP-2	OTP-3	OTP-4	OTP-5	Locked

Figure 1: Memory map showing Capability Container (CC) *Initialised* State

5.5 TLV and NDEF storage location

TLV stands for <Tag>, <Length>, <Value> and is a structure used to introduce the NDEF message location and size within the Type 1 tag, see Figure 1.

Block-1, byte-4 is used as the <Tag>, T=03_h to indicate an NDEF message.

Block-1, byte-5 is used to show the <Length> of the NDEF message. Here L=00_h, to indicate a null message.

A more detailed example with a real NDEF message is given in section 8.

6. Read-only Locking

6.1 Soft read-only

There are actually two levels of controlling the change to read-only. The first level is to change the value of the Read Write Access (RWA) indicator in the capability container from 00_h to $0F_h$. This will indicate to an NFC Device, eg Nokia 6131 NFC phone that the tag is for read-only operation and it will not attempt to over write the data if you try to send new data to the tag. This level is actually reversible given an NFC Device or RFID reader/writer with the correct software and it is perfectly possible to alter the data contents at any time. This “soft” *read-only* is not recommended.

6.2 Hard lock read-only

The most secure method is to physically lock the whole tag memory into ‘read-only’ mode to make all of the tag data incapable of being changed ever again. Firstly, the CC should be programmed so that the RWA = $0F_h$ to show the NDEF message to be *Read-only*. Then additionally all of the physical lock bits of the tag should be set by programming the value FF_h into both of lock bytes called LOCK-0 and LOCK-1 located in block-E, byte-0 and byte-1.

7. Overview of Life-Cycle States

7.1 States

During the life-cycle of the NFC Forum Type 1 tag it can be considered to be in one of three states as follows:

1) *Initialised*
2) *Read/write*
3) *Read-only*

An NFC Forum Device in reader/writer mode will interpret the Type 1 tag to be in one of these states.

The state will be reflected by the contents of the tag.

7.2 Initialised

A Type 1 tag is considered to be in the *Initialised* state when not in the *Read/write* or *Read-only* states.

The tag contains a valid CC and a TLV, which indicates an NDEF message of zero length.

The tag memory map in Figure 1 shows a tag in the *Initialised* state.

7.3 Read/write

A Type 1 tag is considered to be in the *Read/write* state when there is a non-zero length NDEF message present and CC3=00_h.

7.4 Read-only

A Type 1 tag is considered to be in the *Read-only* state when there is a non-zero length NDEF message present and when CC3=$0F_h$.

8. Examples

8.1 Initialised

The tag memory map in Figure 1 has already shown a tag in the *Initialised* state.

8.2 Read/write

The tag memory map in Figure 2 shows a tag containing an NDEF message in the *Read/write* state.

HR0	HR1
11_h	XX_h

EEPROM Memory Map									
Memory Block	Byte-0 **(LSB)**	Byte-1	Byte-2	Byte-3	Byte-4	Byte-5	Byte-6	Byte-7 **(MSB)**	Lock Condition
0	UID-0	UID-1	UID-2	UID-3	UID-4	UID-5	25_h		Locked
1	CC0 (NMN) $=E1_h$	CC1 (Vno) $=10_h$	CC2 (TMS) $=0E_h$	CC3 (RWA) $=00_h$	NDEF Message TLV (Tag) $T=03_h$	NDEF Message TLV (Length) $L=30_h$	NDEF Message TLV (Value) $D1_h$	02_h	
2	$2B_h$	53_h	70_h	91_h	01_h	06_h	54_h	02_h	
3	65_h	$6E_h$	49_h	72_h	74_h	51_h	01_h	$1D_h$	
4	55_h	03_h	$6E_h$	66_h	63_h	64_h	65_h	$6D_h$	
5	$6F_h$	$2E_h$	69_h	$6E_h$	$6E_h$	$6F_h$	76_h	69_h	
6	73_h	69_h	$6F_h$	$6E_h$	$2D_h$	67_h	72_h	$6F_h$	
7	75_h	70_h	$2E_h$	63_h	$6F_h$	$6D_h$	NDEF Message TLV (Tag) $T=FE_h$		
8									
9									
A									
B									
C									
D									Locked
E	LOCK-0 $=01_h$	LOCK-1 $=60_h$	OTP-0	OTP-1	OTP-2	OTP-3	OTP-4	OTP-5	Locked

Figure 2 Memory map showing Example NDEF Message in *Read/Write* State

The <Length> field of the TLV is 30_h, which indicates an NDEF message of 48 Bytes in length.

The NDEF message which is the <Value> part of the TLV structure, starts at block-1, byte-6 and ends at block-7, byte-5.

The TLV with <Tag>, $T=FE_h$ located at block-7, byte-6, indicates the end of NDEF message

storage on this tag.

The Read Write Access (RWA) permission in the CC3 = 00_h indicates that the NDEF message is to be fully read/write.

LOCK-0 = 01_h and LOCK-1 = 06_h, which are the default values for the *Initialised* tag is as required for all read/write data blocks 1 to C_h to remain unlocked and therefore capable of being written.

8.3 Read-only

The tag memory map in Figure 3 shows a tag containing the same NDEF message in the *Read-only* state.

HR0	HR1
11_h	XX_h

EEPROM Memory Map									
Memory Block	Byte-0 **(LSB)**	Byte-1	Byte-2	Byte-3	Byte-4	Byte-5	Byte-6	Byte-7 **(MSB)**	Lock Condition
0	UID-0	UID-1	UID-2	UID-3	UID-4	UID-5	25_h		Locked
1	CC0 (NMN) $=E1_h$	CC1 (Vno) $=10_h$	CC2 (TMS) $=0E_h$	CC3 (RWA) $=0F_h$	NDEF Message TLV (Tag) $T=03_h$	NDEF Message TLV (Length) $L=30_h$	NDEF Message TLV (Value) $D1_h$	02_h	Locked
2	$2B_h$	53_h	70_h	91_h	01_h	06_h	54_h	02_h	Locked
3	65_h	$6E_h$	49_h	72_h	74_h	51_h	01_h	$1D_h$	Locked
4	55_h	03_h	$6E_h$	66_h	63_h	64_h	65_h	$6D_h$	Locked
5	$6F_h$	$2E_h$	69_h	$6E_h$	$6E_h$	$6F_h$	76_h	69_h	Locked
6	73_h	69_h	$6F_h$	$6E_h$	$2D_h$	67_h	72_h	$6F_h$	Locked
7	75_h	70_h	$2E_h$	63_h	$6F_h$	$6D_h$	NDEF Message TLV (Tag) $T=FE_h$		Locked
8									Locked
9									Locked
A									Locked
B									Locked
C									Locked
D									Locked
E	LOCK-0 **$=FF_h$**	LOCK-1 **$=FF_h$**	OTP-0	OTP-1	OTP-2	OTP-3	OTP-4	OTP-5	Locked

Figure 3 Memory map showing Example NDEF Message in *Read-only* State

The Read Write Access (RWA) permission in the CC3 = $0F_h$ indicates that the NDEF message is to be *Read-only*, ie write permission denied.

LOCK-0 = FF_h and LOCK-1 = FF, set the tag such that all blocks are now locked permanently to read-only operation. An NFC device will fail should it attempt to send data to this tag.

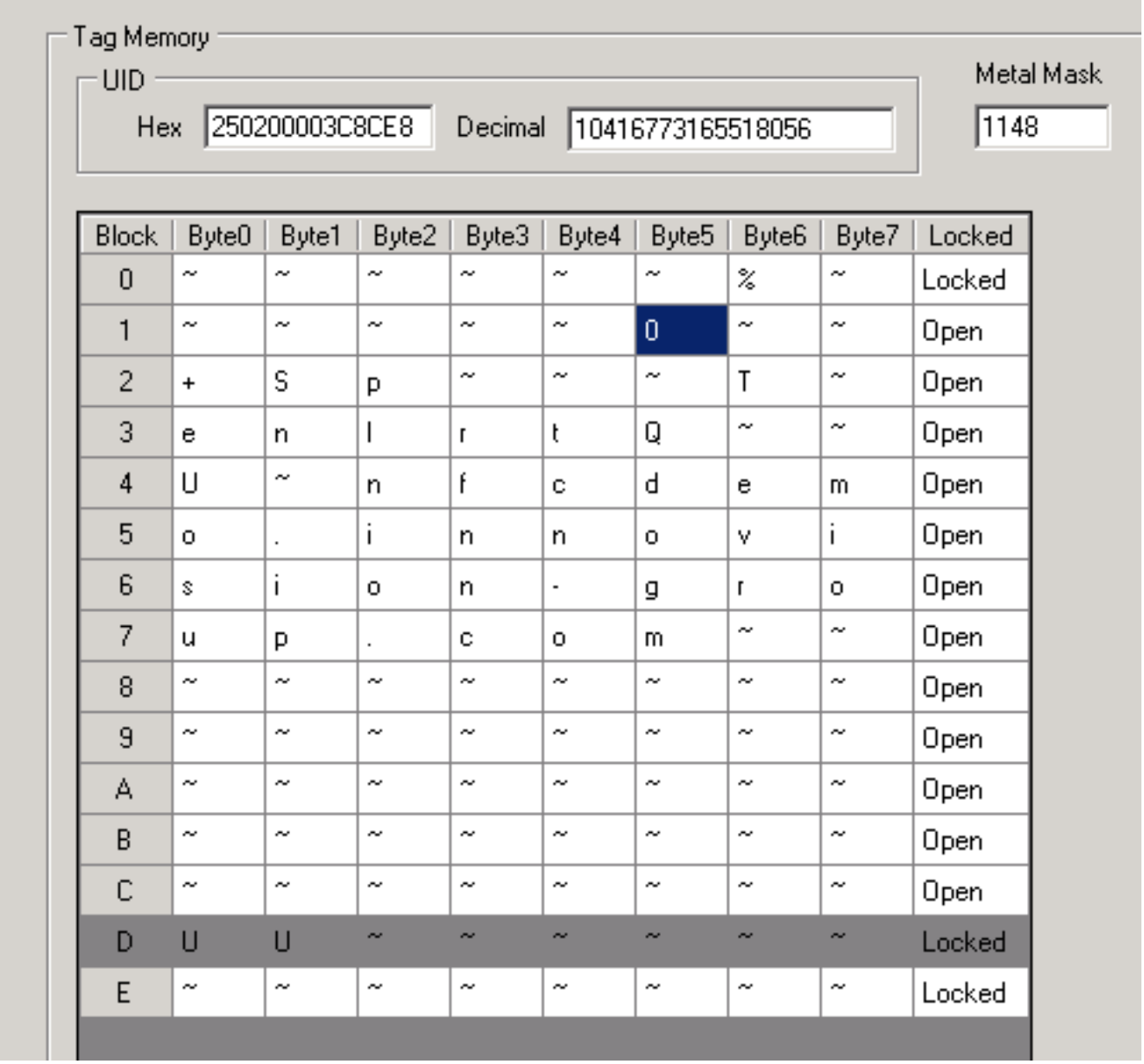
8.4 Example NDEF Message

Figure 4 shows the “ASCII” equivalent of the NDEF message data from in Figure 2.

This example NDEF message is a ‘Bookmark’ for the Innovision NFCdemo website URL address.

It is possible to see the ASCII characters starting from block-4, byte-2.

nfcdemo.innovision-group.com



Block	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Locked
0	~	~	~	~	~	~	%	~	Locked
1	~	~	~	~	~	0	~	~	Open
2	+	S	p	~	~	~	T	~	Open
3	e	n	l	r	t	Q	~	~	Open
4	U	~	n	f	c	d	e	m	Open
5	o	.	i	n	n	o	v	i	Open
6	s	i	o	n	-	g	r	o	Open
7	u	p	.	c	o	m	~	~	Open
8	~	~	~	~	~	~	~	~	Open
9	~	~	~	~	~	~	~	~	Open
A	~	~	~	~	~	~	~	~	Open
B	~	~	~	~	~	~	~	~	Open
C	~	~	~	~	~	~	~	~	Open
D	U	U	~	~	~	~	~	~	Locked
E	~	~	~	~	~	~	~	~	Locked

Figure 4 Example NDEF Message